```
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSS
RRR   RRR        MMM         MMM              SSS
RRR   RRR        MMM         MMM              SSS
RRR   RRR        MMM         MMM              SSS
RRR      RRR     MMM         MMM              SSS
RRR      RRR     MMM         MMM              SSS
RRR      RRR     MMM         MMM              SSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
```

```
_$

Sy
--
NT
NT
NT
NT
NT
NT


NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT

NT

NT
NT
NT
NT
NT
NT
NT



NT
NT
NT
NT
NT
NT
PI
```

```
                                        G 12
**FILE**ID**RMSOREWIN









RRRRRRRR    MM      MM    SSSSSSSS    000000    RRRRRRRR    EEEEEEEEEE  WW      WW    IIIIII    NN      NN
RRRRRRRR    MM      MM    SSSSSSSS    000000    RRRRRRRR    EEEEEEEEEE  WW      WW    IIIIII    NN      NN
RR      RR  MMMM  MMMM  SS          00      00  RR      RR  EE          WW      WW      II      NN      NN
RR      RR  MMMM  MMMM  SS          00      00  RR      RR  EE          WW      WW      II      NN      NN
RR      RR  MM  MM  MM  SS          00    0000  RR      RR  EE          WW      WW      II      NNNN    NN
RR      RR  MM  MM  MM  SS          00    0000  RR      RR  EE          WW      WW      II      NNNN    NN
RRRRRRRR    MM      MM    SSSSSS    00  00  00  RRRRRRRR    EEEEEEEE    WW      WW      II      NN  NN  NN
RRRRRRRR    MM      MM    SSSSSS    00  00  00  RRRRRRRR    EEEEEEEE    WW      WW      II      NN  NN  NN
RR  RR      MM      MM          SS  0000    00  RR  RR      EE          WW  WW  WW      II      NN    NNNN
RR  RR      MM      MM          SS  0000    00  RR  RR      EE          WW  WW  WW      II      NN    NNNN
RR    RR    MM      MM          SS  00      00  RR    RR    EE          WWWW  WWWW      II      NN      NN      ....
RR    RR    MM      MM          SS  00      00  RR    RR    EE          WWWW  WWWW      II      NN      NN      ....
RR      RR  MM      MM  SSSSSSSS      000000    RR      RR  EEEEEEEEEE  WW      WW    IIIIII    NN      NN      ....
RR      RR  MM      MM  SSSSSSSS      000000    RR      RR  EEEEEEEEEE  WW      WW    IIIIII    NN      NN      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
    (2)        76          DECLARATIONS
    (3)       108          RMS$REWIND - $REWIND FILE ROUTINE
```

```
0000        $BEGIN  RMSOREWIN,000,RM$RMS,<FILE REWIND OPERATION>
0000
0000  ;
0000  ;*************************************************************************
0000  ;*                                                                       *
0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.               *
0000  ;*  ALL RIGHTS RESERVED.                                                 *
0000  ;*                                                                       *
0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0000  ;*  TRANSFERRED.                                                         *
0000  ;*                                                                       *
0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0000  ;*  CORPORATION.                                                         *
0000  ;*                                                                       *
0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.               *
0000  ;*                                                                       *
0000  ;*                                                                       *
0000  ;*************************************************************************
0000  ;
0000  ;++
0000  ; Facility: rms32
0000  ;
0000  ; Abstract:
0000  ;                 this routine is the highest level control
0000  ;                 routine to perform the $rewind function.
0000  ;
0000  ; Environment:
0000  ;                 star processor running starlet exec.
0000  ;
0000  ; Author: l f laverdure,           creation date: 24-OCT-1977
0000  ;
0000  ; Modified By:
0000  ;
0000  ;       V03-004 KPL0001         Peter Lieberwirth       20-Jun-1983
0000  ;               Change some references to JNLFLG to JNLFLG2.
0000  ;
0000  ;       V03-003 JWH0182         Jeffrey W. Horn         10-Feb-1983
0000  ;               If we are doing 'fake' record locking and we're in a
0000  ;               recovery unit, then make sure we make call to RM$UNLOCKALL,
0000  ;               regardless of the state of IFB$V_NORECLK.
0000  ;
0000  ;       V03-002 KBT0190         Keith B. Thompson       23-Aug-1982
0000  ;               Reorganize psects and rename entry point to single '$'
0000  ;
0000  ;       V03-001 CDS0002         C Saether       17-Feb-1982
0000  ;               Don't clear IRB$V_GBLBUFF on rewind.
0000  ;
0000  ;       V02-014 CDS0001         C Saether       03-Aug-1981
0000  ;               Remove SQO check around PPF check.
0000  ;
```

```
0000   ;            V02-013 KPL0001          P Lieberwirth    39-Jul-1981
0000   ;                    Change entry point to $$ form so we can get here via
0000   ;                    RMS0BRNCH.
0000   ;
0000   ;            V02-012 KRM0002          K R Malik        09-DEC-1980
0000   ;                    modify BKPMASK to prevent clearing IRB$V_DAP_CON.
0000   ;
0000   ;            V02-011 REFORMAT         P S Knibbe       25-Jul-1980
0000   ;
0000   ;            V010    JAK0020          J A Krycka       05-SEP-1979      09:00
0000   ;                    release 2.0 work.
0000   ;
0000   ;            V008    RAN0002          R A Newell       06-SEP-1978      08:53
0000   ;                    rms32 isam modification.
0000   ;
0000   ;--
0000
```

```
                  0000          .SBTTL  DECLARATIONS
                  0000
                  0000  ;
                  0000  ; Include Files:
                  0000  ;
                  0000
                  0000  ;
                  0000  ; Macros:
                  0000  ;
                  0000
                  0000          $DEVDEF
                  0000          $IFBDEF
                  0000          $IRBDEF
                  0000          $RMSDEF
                  0000
                  0000  ;
                  0000  ; Equated Symbols:
                  0000  ;
                  0000
        00000020  0000          BKP=IRB$L_BKPBITS*8               ; bit offset to bookkeeping bits
                  0000          BKPMASK=^C<<1@<IRB$V_BUSY-BKP>>!-
                  0000                  <1@<IRB$V_ASYNC-BKP>>!-
                  0000                  <1@<IRB$V_ASYNCWAIT-BKP>>!-
                  0000                  <1@<IRB$V_RAHWBH-BKP>>!-
                  0000                  <1@<IRB$V_GBLBUFF-BKP>>!-
        EFBFFBE6  0000                  <1@<IRB$V_DAP_CONN-BKP>>>         ; mask to reset bits
                  0000
                  0000  ;
                  0000  ; Own Storage:
                  0000  ;
                  0000
```

```
0000            .SBTTL  RMS$REWIND - $REWIND FILE ROUTINE
0000
0000    ;++
0000    ; RMS$REWIND - Rewind file
0000    ;
0000    ; this routine performs common rab function setup followed by an
0000    ; implicit $free and $flush and then by dispatch to organization-dependent
0000    ; $rewind code
0000    ;
0000    ; Calling sequence:
0000    ;
0000    ;       entered from exec as a result of user's calling sys$rewind
0000    ;       (e.g., by using the $rewind macro)
0000    ;
0000    ; Input Parameters:
0000    ;
0000    ;       ap      user's argument list addr
0000    ;
0000    ; Implicit Inputs:
0000    ;
0000    ;       the contents of the rab and related irab and ifab.
0000    ;
0000    ; Output Parameters:
0000    ;
0000    ;       r1      destroyed
0000    ;       r0      status code
0000    ;
0000    ; Implicit Outputs:
0000    ;
0000    ;       all locked records for the stream are freed, and the stream is
0000    ;       repositioned at the start of the file.  the file attributes are also
0000    ;       rewritten.
0000    ;       for the sequential org., all bdb's and buffers are released.
0000    ;       rab$l_sts and rab$l_stv reflect the status of the rewind.
0000    ;
0000    ;       a completion ast is queued if specified in the user arglist.
0000    ;
0000    ; Completion Codes:
0000    ;
0000    ;       standard rms (see functional spec for list).
0000    ;
0000    ; Side Effects:
0000    ;
0000    ;       none
0000    ;
0000    ;--
0000
```

```
                     0000   156              $ENTRY  RMS$REWIND
                     0000   157              $TSTPT  REWIND
                     0006   158              $RABSET                               ; do common setup with no specific checks
                     000A   159
                     000A   160 ;
                     000A   161 ; returns to user on error
                     000A   162 ;
                     000A   163
  63 6A    3E    E0  000A   164              BBS     #IFB$V_DAP,(R10),NTRWND ; branch if network operation
           22    E0  000E   165              BBS     #IRB$V_PPF_IMAGE,-
        6B 69        0010   166                      (R9),ERRIOP             ; branch if indirect ppf
           0E    E1  0012   167              BBC     #DEV$V_FOD,-
           6A        0014   168                      IFB$L_PRIM_DEV(R10),-
           67        0015   169                      ERRIOP                  ; or if unit rec device
           02    E1  0016   170              BBC     #IFB$V_RUP,-
  06 00A2  CA        0018   171                      IFB$B_JNLFLG2(R10),10$  ; branch if not in RU
           03    E0  001C   172              BBS     #IFB$V_RU_RLK,-
  04 00A2  C9        001E   173                      IFB$B_JNLFLG2(R9),20$   ; branch if 'fake' record locking
  03 6A    33    E0  0022   174 10$:         BBS     #IFB$V_NORECLK,(R10),30$; omit record unlock if no locking
        FFD7'    30  0026   175 20$:         BSBW    RM$UNLOCKALL            ; free any locked records
                     0029   176                                              ; (ignore possible error)
                     0029   177 30$:
                     0029   178
                     0029   179 ;
                     0029   180 ; dispatch to org-dependent code
                     0029   181 ;
                     0029   182
                     0029   183              CASE    TYPE=B,-
                     0029   184                      SRC=IFB$B_ORGCASE(R10),-; and dispatch
                     0029   185                      DISPLIST=<RWSEQ,RWREL,RWIDX>
                     0034   186
                     0034   187 ;
                     0034   188 ;*********************************************************************************
                     0034   189 ;
                     0034   190 ;                indexed specific rewind
                     0034   191 ;
                     0034   192 ;**********************************************************************************
                     0034   193 ;
                     0034   194
  00000000'EF    16  0034   195 RWIDX:       JSB     RM$REWIND3              ; do indexed specific rewind
           27    11  003A   196              BRB     DONE
                     003C   197
                     003C   198 ;*************************************************************************************
                     003C   199 ;
                     003C   200 ;  sequential file org rewind
                     003C   201 ;
                     003C   202 ;*************************************************************************************
                     003C   203
        FFC1'    30  003C   204 RWSEQ:       BSBW    RM$FLUSH                ; write and release any bdbs
                     003F   205                                              ; rewriting file attributes
     2C 50       E9  003F   206              BLBC    R0,EXIT                 ; get out on error
     62 A9       B4  0042   207              CLRW    IRB$W_CSIZ(R9)          ; reset to no current record
           05    E1  0045   208              BBC     #DEV$V_SQD,-
           6A        0047   209                      IFB$L_PRIM_DEV(R10),-
           0D        0048   210                      RWREL                   ; branch if not magtape
                     0049   211
                     0049   212 ;
```

```
                          0049   213 ;   magtape support
                          0049   214 ;
                          0049   215
        74 AA    01   CE  0049   216         MNEGL   #1,IFB$L_EBK(R10)           ; set eof block to max value
           5C AA      B4  004D   217         CLRW    IFB$W_FFB(R10)              ; reset eof offset
            FFAD'     30  0050   218         BSBW    RM$REWIND_MT                ; do rewind
           18 50      E9  0053   219         BLBC    R0,EXIT                     ; get out on error
                          0056   220                                             ; else fall thru
                          0056   221
                          0056   222 ;*****************************************************************************
                          0056   223 ;
                          0056   224 ;   relative file org rewind
                          0056   225 ;
                          0056   226 ;*****************************************************************************
                          0056   227
                          0056   228         ASSUME  IRB$L_NRP_OFF EQ IRB$L_NRP_VBN+4
                          0056   229
           40 A9      7C  0056   230 RWREL:  CLRQ    IRB$L_NRP_VBN(R9)           ; reset nrp
              18      E0  0059   231         BBS     #DEV$V_FOR,-
            03 6A         005B   232                 IFB$L_PRIM_DEV(R10),10$ ; branch if non-file-struc.
           40 A9      96  005D   233         INCB    IRB$L_NRP_VBN(R9)           ; nrp = vbn 1
                          0060   234
                          0060   235 10$:    ASSUME  IRB$L_RP_OFF EQ IRB$L_RP_VBN+4
                          0060   236
           48 A9      7C  0060   237         CLRQ    IRB$L_RP_VBN(R9)
                          0063   238
                          0063   239 ;
                          0063   240 ;   clear all irab bookkeeping bits except busy, async, asyncwait, rahwbh
                          0063   241 ;   gblbuff & dap_con.
                          0063   242 ;
                          0063   243
04 A9   EFBFFBE6 8F   CA  0063   244 DONE:   BICL    #BKPMASK,IRB$L_BKPBITS(R9)
                          006B   245         RMSSUC
            FF8F'     31  006E   246 EXIT:   BRW     RM$EXRMS                    ; return to caller
                          0071   247
                          0071   248 ;++
                          0071   249 ;
                          0071   250 ;   perform network rewind function
                          0071   251 ;
                          0071   252 ;--
                          0071   253
                          0071   254 NTRWND:
            FF8C'     30  0071   255         BSBW    NT$REWIND                   ; rewind the file
           F7 50      E9  0074   256         BLBC    R0,EXIT                     ; branch on error
        40 A9    01   D0  0077   257         MOVL    #1,IRB$L_NRP_VBN(R9)        ; reset next block pointer
              E6      11  007B   258         BRB     DONE                        ; rejoin mainline
                          007D   259
                          007D   260 ;
                          007D   261 ;   handle error
                          007D   262 ;
                          007D   263
                          007D   264 ERRIOP:
                          007D   265         RMSERR  IOP                         ; can't rewind unit rec dev. or
                          0082   266                                             ; process perm. file if indirectly
                          0082   267                                             ; connected
              EA      11  0082   268         BRB     EXIT
                          0084   269
```

```
               0084   270          .END
```

```
$$.PSECT_EP                      = 00000000
$$RMSTEST                        = 0000001A
$$RMS_PBUGCHK                    = 00000010
$$RMS_TBUGCHK                    = 00000008
$$RMS_UMODE                      = 00000004
BKP                              = 00000020
BKPMASK                          = EFBFFBE6
DEV$V_FOD                        = 0000000E
DEV$V_FOR                        = 00000018
DEV$V_SQD                        = 00000005
DONE                               00000063 R      01
ERRIOP                             0000007D R      01
EXIT                               0000006E R      01
IFB$B_JNLFLG2                    = 000000A2
IFB$B_ORGCASE                    = 00000023
IFB$L_EBK                        = 00000074
IFB$L_PRIM_DEV                   = 00000000
IFB$V_DAP                        = 0000003E
IFB$V_NORECLK                    = 00000033
IFB$V_RUP                        = 00000002
IFB$V_RU_RLK                     = 00000003
IFB$W_FFB                        = 0000005C
IRB$L_BKPBITS                    = 00000004
IRB$L_NRP_OFF                    = 00000044
IRB$L_NRP_VBN                    = 00000040
IRB$L_RP_OFF                     = 0000004C
IRB$L_RP_VBN                     = 00000048
IRB$V_ASYNC                      = 00000023
IRB$V_ASYNCWAIT                  = 00000024
IRB$V_BUSY                       = 00000020
IRB$V_DAP_CONN                   = 0000003C
IRB$V_GBLBUFF                    = 00000036
IRB$V_PPF_IMAGE                  = 00000022
IRB$V_RAHWBH                     = 0000002A
IRB$W_CSIZ                       = 00000062
NT$REWIND                          ********   X    01
NTRWND                             00000071 R      01
PIO$A_TRACE                        ********   X    01
RM$EXRMS                           ********   X    01
RM$FLUSH                           ********   X    01
RM$REWIND3                         ********   X    01
RM$REWIND_MT                       ********   X    01
RM$RSET                            ********   X    01
RM$UNLOCKALL                       ********   X    01
RMS$REWIND                       = FFFFFFFE RG     01
RMS$_IOP                         = 00018574
RWIDX                              00000034 R      01
RWREL                              00000056 R      01
RWSEQ                              0000003C R      01
TPT$L_REWIND                       ********   X    01
```

```
                                        +----------------+
                                        ! Psect synopsis !
                                        +----------------+

PSECT name                      Allocation           PSECT No.  Attributes
----------                      ----------           ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (   0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
RMS$RMS                         00000084  (   132.)  01 (   1.)    PIC    USR    CON    REL    GBL NOSHR   EXE   RD   NOWRT NOVEC BYTE
$ABS$                           00000000  (     0.)  02 (   2.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                     +------------------------+
                                     ! Performance indicators !
                                     +------------------------+

Phase                     Page faults     CPU Time       Elapsed Time
-----                     -----------     --------       ------------
Initialization                     29     00:00:00.09    00:00:01.07
Command processing                110     00:00:00.79    00:00:04.22
Pass 1                            245     00:00:06.52    00:00:15.80
Symbol table sort                   0     00:00:00.79    00:00:00.92
Pass 2                             59     00:00:01.34    00:00:02.60
Symbol table output                 8     00:00:00.08    00:00:00.15
Psect synopsis output               2     00:00:00.01    00:00:00.04
Cross-reference output              0     00:00:00.00    00:00:00.00
Assembler run totals              455     00:00:09.62    00:00:24.80

The working set limit was 1350 pages.
34665 bytes (68 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 678 non-local and 6 local symbols.
270 source lines were read in Pass 1, producing 13 object records in Pass 2.
21 pages of virtual memory were used to define 20 macros.

                                    +----------------------------+
                                    ! Macro library statistics !
                                    +----------------------------+

Macro library name                           Macros defined
------------------                           --------------
_$255$DUA28:[RMS.OBJ]RMS.MLB;1                         10
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                          1
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       5
TOTALS (all libraries)                                 16

799 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:RMSOREWIN/OBJ=OBJ$:RMSOREWIN MSRC$:RMSOREWIN/UPDATE=(ENH$:RMSOREWIN)+EXECML$/LIB+LIB$:RMS/LIB
```

0330 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY